

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-256064

(43)Date of publication of application : 21.09.2001

(51)Int.Cl.

G06F 9/46
G06F 15/18

(21)Application number : 2000-066405

(71)Applicant : MITSUBISHI ELECTRIC CORP

(22)Date of filing : 10.03.2000

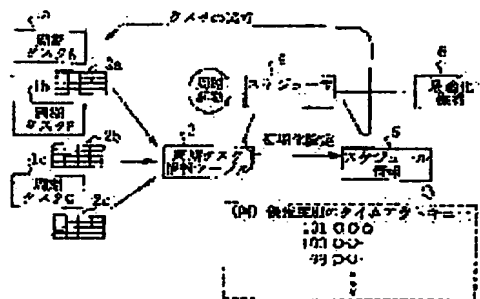
(72)Inventor : KAWAKAMI TAKESHI

(54) OPTIMIZATION SCHEDULING SYSTEM OF PLURAL PERIODIC PERFORMANCE TASKS

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a scheduling system to enable optimization so that width of shift of intervals of performance periods of plural periodic tasks to perform a constant periodic processing is minimized.

SOLUTION: The optimization of scheduling to enable performance of the plural periodic tasks to perform the constant periodic processing is performed at constant periodic intervals by optimizing a performance schedule to perform the plural periodic tasks 1 by optimization mechanism 6 based on intrinsic information by every one of the plural periodic tasks 1 to perform the constant periodic processing and correcting the performance schedule of the plural preset periodic tasks 1 according to the optimized performance schedule by a scheduler 4.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

THIS PAGE BLANK (USPTO)

(A) 英國華平公司 (B)

(11) 特許出願公開番号
特開2001-256064
(P2001-256064A)

（43）公開日 平成13年9月21日（2001.9.21）

識別記号
340
550

テ-77-1⁰(~~秘~~考) 5B098

審査請求 未請求 請求項の数11 O.L (全22頁)

特選2000-66405(P2000-66405)
平成12年3月10日(2000.3.10)

(71) 出票人 000006013

東京都千代田区丸の内二丁目2番3号

(72) 发明者 川上 武

荏電機株式会社内

(74) 代理人 100102439

弁理士 宮田 金雄 (外1名)

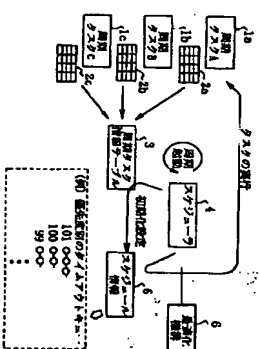
FT-7L(参考) SB098 FT04 GA04 GA08 CO05

(54) 【発明の名称】 複数周期実行タスクの最適化スケジューリング方式

(57) 【要約】

【課題】 定周期処理を実行する複数の周期タスクの実行周期の間隔が極小値となるような最適化が可能となるスケジューリング方式を提供する。

【解決手段】 定期利息返を受ける債権の定期タスク1毎の留付情報に基づいて、最速に期限6で債権の定期タスク1を受ける実行スケジュールの最速化を行い、スケジュール4でこの最速化された実行スケジュールに従って、予め設定された債権の定期タスク1の実行スケジュールを修正することにより、定期利息返を受ける債権の定期タスクを一定の定期間隔で実行できるとするスケジュール5の最速化を行うことができる。



【特許請求の範囲】

【請求項1】 定周期処理を実行する複数の周期タスク毎の固有情報を格納するテーブルと、

上座アージュルの上座僧が情報に基づいて、上座僧数の局
部タスクが実行する定期処理の一周毎の周期間隔を
一定にする実行スケジュールの最適化が必要か否かの判
定を行う、最適化が必要なときには実行スケジュールの
最適化を行う最適化手段と、

上記最適化手段により最適化された実行スケジュールに基づいて、予め設定された上記複数の同期タスクの実行スケジュールを修正するスケジューラと、を備えたことを特徴とする複数の同期実行タスクの最適化スケジューリング方式。

【請求項2】 上記固有情報は、上記複数の周期タスク毎に予め設定されている上記複数の周期タスクを実行する第1の周期時間情報と、上記複数の周期タスクを実行するときに生じる上記第1の周期時間情報の変化量を表す第2の周期時間情報と、上記複数の周期タスクが前回実行されたときの第2の周期時間情報とを備え、

上記最速に手続は、上記問題時刻情報値が0より0.0以外の場合を判断し、上記問題時刻情報が0以外の時には上記第2の局所時間情報と上記第1の局所時間情報とが一致しているか否かを判定し、一致していないときには、上記問題時刻情報に基づいて上記第1の局所時間情報の変更を行うことと特徴とする請求項1記載の項数西暦実行プログラム。なお、最速にスケジューリング方式。

【請求項3】上記固有情報は、複数の問題時間情報を備え、

上記最適化手段は、上記複数の問題時間情報のいずれかに基づいて上記第1の周期時間情報の変更を行うことを特徴とする請求項2記載の複数周期実行タスクの最適化スケジューリング方式。

【請求項4】 上記通信情報は、上記装置の同期スロットが前4回実行されたときの第3の同期時間情報とを有し、
上記装置は、上記第2の同期時間情報と上記第3の同期時間情報との比較を行い、この比較結果に基づいて上記同期時間情報を決定し、この決定された同期時間情報に基づいて、上記第1の同期時間情報の変更を行うことを特徴とする請求項2記載の装置、同期実行タスクの最適化スケジューリング方式。

【請求項5】 上記第1の時刻同期情報と上記第2の時刻同期情報を力する第1のニューター群と、次の実行プログラムジェネレータで決定した要となる2つの時刻同期情報出力力ニューター群のニューター群と、上記第1のニューター群と第2のニューター群と、上記第1の時刻同期情報と上記第2の時刻同期情報とに基づいて第3のニューター群に出力される第3の時刻同期情報とを決定するものための処理を出力する第3のニューター群と、第1のニューター群で出力される上記第1の時刻同期情報と上記第2の時刻同期情報と、上記第3のニューター群で出力される情報の一部とを力する第4のニューター群で出力される情報の一部とを力する第5のニューター群を有する。

ニューロン群と、第4のニューロン群から第3のニューロン群に出力される上記第2の問題時間情報を決定するための情報を導き出す第4のニューロン群と、から構成されるニューラルネットワークを備え、

上記最適化手段は、上記ニューラルネットワークにより出力された次の実行スケジュール設定時に必要となる第2の問題時間情報に基づいて上記第1の周期時間情報の変更を行うことを特徴とする請求項2記載の装置、および実行タスクの最適化スケジューリング方式。

【請求項6】 上記複数の周期タスクの実行状況を確する利用者により命令が入力される命令入力手段を備え、

上記記載の手続きは、上記第1の周期時間情報の変更を行う
うが、右を示すラックを格納した第1のメモリを備え、
上記命令実行手段からの命令に基づいて上記ラックが変更
されたら、この変更されたラックに基づいて、上記記憶
の周期スロットの実行スケジュールの最適化を行うことを
特徴とする請求項1記載の複数周期実行タスクの最適化
スケジューリング方式。

【請求項7】 上記最適化手段は、上記複数の同期タスク毎に同期タスクの最適化が全て終了する最速に実行可能な情報を格納した第2のメモリを備え、この最速に実行可能な情報に基づいて上記複数の同期タスクの実行スケジュールの最速化を行うことを特徴とする請求項1記載の複数同期実行タスクの最速化スケジューリング方式。

【請求項8】 上記複数の周知タスクの実行スケジュールを、上記複数の周知タスクの実行状況を通知する利用者からの指示に基づいて最適化するアプリケーションソフトウェアを備えたことを特徴とする請求項1記載の最適化手段を備えたことを特徴とする請求項1記載の最適化手段の実行タスクの最適化スケジュール方式。

【請求項9】 上記アプリケーション便最適化手段は上記データと、上記最適化手段と、上記スケジューと、上記フラグを格納する第1のメモリと、上記最適化実行時情報を格納した第2のメモリと同一機能を

する、それぞれ擬似テラブルと、擬似最適化手段と、
似ステジュラと、上記テラブルを格納する第1の擬
メモリと、擬似最適化実行時間情報を格納する第2の擬
メモリとを備えたことを特徴とする請求項8記載の
数値最適化システムの最適化ステジュラリソング方式。

【例事項10】 上記強制手段は、上記スケジュールの実施を一時停止するスケジュール強制停止手段と、上記スケジュール強制停止手段によるスケジュールの一時停止中、上記スケジューリング制御装置が最優先された上記実行スケジュールを上記スケジュールが収容された実行スケジュールと変換するスケジュール内容変更手段を備えたことを特徴とする請求項1記載の装置である。

【請求項11】 上記アブリケーション側最適化手段は、上記複数の周期タスクに新たな周期タスクを追加したときに、この新規周期タスクの固有情報を格納し

う。この後、スケジューラ4は、自分自身が周期的に起動され、スケジュール情報を参照し、タイムアウトした同期タスクがあればこの同期タスクの実行を行うことを繰り返す。

[illegible]

[0026] ステージS102でアイドル時に同期タスクが無い場合は、処理を行わずにスLEEPするが、ステージS102でアイドル時にこの同期タスクが存在しなかった場合には、ステージS103でこの同期タスクの実行を行う。実行が完了した後は、この同期タスクの間隔を計測して再度実行できるようにスケジュールングを実行する。この間、スケジューラは、実行が完了した同期タスクを次の同期時刻に実行できるより、再スケジュールングを行う。

【0027】即ち、スラッシュ4は44447101とした
周知タプルである。これを解いて、再スラッシュ510
4であるが、本実施の形態では、スラッシュ510
4で再スラッシュ107する際に、最上位処理が戻で戻で
あるもの判定を行い、スラッシュ106で最上位が必要
と判定された時には、スラッシュ106で最上位の処理
を行い、スラッシュ107で再スラッシュ107を行
う。ここで、図3の例で言えば、スラッシュ5104～
スラッシュ106の処理は、最上位処理も行われる。
要である。

【0028】次に、最適化の処理方法について説明する。図1は、最適化は繰64回行う最適化の処理処理の流れを示すフローチャートである。このフローチャートは、図2の最端で描かれたサブプログラム104〜サブプログラム109の処理をさらに詳しく説明したものである。最適化の処理は、まずこの初期スタックが最適化のために用いられる問題を通して実行どのような最適化をする。この方法と類似して、サブプログラム111で問題空間の0以外の値に設定されているのが、サブプログラム112で各問題スタック毎に設定されている問題空間が0であるか0以外の値があるかを調べる。

[0029] ステックS112で処理時間が0の場合には、最終的な処理を可も行わず、スタックS116で処理時間を初期値のまま設定し、処理を終える。一方、スタックS112で処理時間が0以外であった場合には、ステップS113で前回実行された時から今回の実行ま

での配座（時刻）、即ち周期グラフ1毎の固有情報2に
あらかじめ設定されている前回の周期情報と、周期グラフ
1毎の固有情報2にあらかじめ設定されている（要求
されている）周期時間12との比較を行い、ステップS
114にて一致していることが確認された場合には、特に
最適化を行う必要がないため、ステップS116で周期
情報を初期値のまま設定する。

【0030】スラッシュ14で表していることが
海蔵された場合には、スラッシュ15で時間間隔を変
更して設定を行う。変更の方法としては時間区間の値を
このまま時間間隔に加算もしくは減算し、新しい時間時
間でのスラッシュ15を行う方法、または、0以
上、時間区間の値を乱数で定め、この値を時間区間
に加算し、次のスラッシュ15を行う方法などが考え
られる。

【0001】最面化処理は、この再スキャンゾーンを
行う時に発生する、一例を用いて説明すると、周知
スキャン情報からなる中に周知スキャンがアノードした
たこととを判断するのに必要スキャンがアノードした
値を保持するようなスキャンゾーンゾニアムでは、周
知スキャン1がよりよく実行を要したならば、周
知スキャン1bのタイムアウトゾニアムに周知スキャン1b
の周知時間の値をのこすまでか、指定された周知時間
後に再度実行できるようにするものであるが、周知スキャン
が周知時間と異なった場合には、次の周知時間に変更
して設定する。変更する方法としては、周知時間と最大
値を加算する方法や、スキャン1がより速い最も小
さいタイムアウト値を用い、

【10032】当然のことながら、今回の所期時間と変更した所期スパンの差が、この次に要するときに再び受けるべき変動の幅を決定する。変動の幅が所期時間と異なるので、一度変更を行った場合には差が生ずるようになる。前回に行われた変更を考慮しなければならない、例えば、前回の所期時間を獲得するのに、現時点での所期を認識し、前回の時との差分を定める方が考えられるが、容易に行えるように、変更を行った時の時刻には、この変更を加味した値を保持しておかなければならぬ。

【0030】以上のように、本発明の所望によれば、前回の要因分析機能が、所望されていくと進んだ周期タスクに対して試験を行うことにより、要因分析を最適化する。そして、定期処理処理を実行する複数の周期タスク毎に実行されることには、どの周期タスクも所期タスク毎に設定された一定の周期で定期処理処理が実行されるようなスケジューリングの最適化を行うことができる。

【0034】実施の形態2。以下、実施の形態2を図を参照して説明する。本実施の形態は、問題のために変更することができ、問題の値を複数保持し、これらの問題を選択使用すること、周期的に実行すること、問題を求める速度を向上、

また、周所解（ローカルミニマ）に陥った場合に脱出可能である場合について説明する。

【0035】図5は、固有周期2.4～2.6秒の内部構造図である。図5は、固有周期2.4を1を示す説明図である。変換の形態1との違いは、図5における固有周期1.6と前回の固有周期1.7が複数値が用意されていることである。図5では、問題時刻と前回固有周期時刻がそれぞれ固有周期テーブル2.1、前固有周期テーブル2.2になっている。

【0036】次に、本実施の形態では、どのようにして最適化が行われるのかについて説明する。例えば、図5の時間テーブル2に照準時間を10と20の2つ(21 a、21 b)、前回の時間テーブル22に照準時間を10値(22 a、22 b、...)持つ周期タスクdがある。周期タスクdの周期時間を30とし、この値に周期時間が100の周期タスクa、b、そして周期時間20の周期タスクcがある。

【0033】全ての周期タスクの優先度が同一であると
し、従来の方法でスケジューリングが行われると、図6
に示すように、周期時間の短いもの(例、即ち、周期タ
スクa、b、c、dの順にC/Pが実行)が優先される。す
るべし、周期タスクa～cの周期時間は常に同一であり、
維持されている値になるのであるが、一番周期時間の長
い周期タスクdは、遅れが生じてしまう。

【0038】このような場合において、受給者の形直で受領した形式を使用した場合に、受給者の値の改変を許してしまふと、上記のぶれを是正することができない場合がある。上記の通りで言えば、受給者の値が1.0であるとして、いつまでも替へても妨がはない。そこで、受給時間を数値用ゐることにより、局所からの取出しの際、例えば、受給時間が変異し、1.0と聞かす、全ての受給時間を比較すると、常に1.0のぶれ幅が存在するような場合には(図3の受給時間2.1 aの「1.0」を使用)、数値化以降の中で、受給時間が受当な値ではないと判断し、図3の受給時間2.1 bの「2」に切り替えて値を捨てる。このように受給時間2.1 bになり、図4タマ4の局所時間は解きあがっていることになる。

【0039】即ち、両端タヌA1aを基型として、両端タヌA1aの2位塩基に両端タヌC1bが開始される、次に両端タヌB1bの2位塩基に両端タヌC1cが開始される、次に両端タヌA1aの3位塩基に両端タヌB1bが開始される場合において、再度両端タヌC1aから両端タヌB1bまでの両端タヌC1cの移行が行われる際、両端タヌA1a～両端タヌC1cの移行時のタヌインが台、両端タヌC1cが台が生じることがない。

【0040】以上のように、本実施の形態によれば、前回の実周期時間の情報を複数保有し、また、経過時間を

(6) 特開2001-256064

[illegible]

【0042】本発明の形態の構成は、実施の形態2とはほぼ同様であり、相違点は問題時間テーブルが最大問題時間と、現在使用している問題時間の2つだけしかない、図5で説明すると、問題時間テーブル中の21aと21bのみを使用する。ここでは、21aは最大問題時間、21bは現在使用している問題時間を表している。

【0043】処理の基本的な流れは要約の形番1と同じであり、特記点は、図4のフローチャート中のステップS115で「同期時間を変更して設定」を行う際に、この数値を可変にした点である。例えば、前回の同期時間と、このまた前である前々回の同期時間とを比較し、この式によって、次の同期時間を設定する方法である。

[illegible]

[00046] ステップS122で、上図の比較を行った結果、前回の方が前回より大きなため、次の周期時間T_nに追加する調整時間は、ステップS123で前回使用した調整時間の2倍にする。ただし、値は調整時間の最大値を超えないようにする。一回目では、前回の調整時間は実質「0」なのであるが、一回目で最大の調整時間に達しない限り、次回も最大値が定まれる。

【0046】また、二回目に係るにおいて、前回＝6
0)、前々回は170)、前回の履歴時間は110)であ
った場合には、ステップS124で、次の履歴時間
は前回の半分に設定、つまり15)にする。なお、二回
目に係るにおいて、前回、前々回とも180)、前回の履歴
時間は12)であった場合には、局所係数にまっぴい

(7)

る可能性が高い。また、スケジュールS126で、次の調整時間を最大値、つまり1.0にする。

[0047] このように、本例では、調整を可変する手段として単調に2倍、1.2倍を行ったが、この可変する手段を他に採るかしくは調整を用意し、これにより実現することも可能である。

[0048] 以上のように、本実施の形態によれば、前回の調整期間の情報を保持せずし、また、調整期間の最大値と、現在の値を持ち、また、次の調整期間を可変に設定することによって、最適化を行う際に局所解に陥ってしまうのを防ぎ、各調整タスクの実行間隔を最適化することができ、定期調整と実行する調整の調整タスクに設定された一定の周期で定期調整が実行できるようスケジュールの最適化を行うことができる。

[0049] 実施の形態4、以下、実施の形態4を図を参照して説明する。実施の形態は、周期の決め幅を調整する際に使用する調整期間の値を決定するのにニューラルネットワークを用いることを特徴とするスケジュール方式である。実施の形態3では、次の調整期間を可変設定したのであるが、本実施の形態では、この可変設定をニューラルネットワークで自動化し、より早く最適解に収束させるものである。

[0050] 本実施の形態の構成は、実施の形態2とはほぼ同様である。相違点は調整タスクの固有情報として新たな情報を保持している点である。すなわち、新たな情報に保持された固有情報2 a～2 cの内部構造（ここでは調整タスク1 aの固有情報2 a）を示す説明図である。

[0051] 本実施の形態で新たに追加される新規の情報は、ニューラルネットワークの層数30であり、この中にはニューラルネットワークを構成している各ニューロンが保持している情報を保持している。例えば、31は第一のニューロンに貯る情報であり、ニューロンの入力値31 a、ニューロンの出力値31 b、ニューロンの結合情報31 c、出力層数31 d、及び他のニューロンの結合情報31 eを有する。このように、保持するニューロンの数分だけ情報を構築し、システム初期化時にはニューロンの結合処理を行う。

[0052] 以上11は、ニューラルネットワークの形態を示した図である。11.0に示すように、一部のニューロンからの出力を再帰的に入力値として使用する（ニューロン型のニューラルネットワークが用いられる（ニューロンに相当する調整タスク相互の間に期間のある結合をもたせた構成であるホップフィールド型である）。

用）とを合わせたものを入力する。真ん中のニューロン群43は中間層、その上はニューロン群44が出力層である。ニューロン群44の一部のニューロンは、この出力値をそのままニューロン群42にフィードバックさせる。一番上のニューロン群45は出力値であり、次のスケジュールに必要となる調整期間が出力される。

[0054] 各層のニューロン数はあまりに多いと計算時間のオーバーヘッドがあまり大きいため、できる限り最小限にとどめておく必要がある。例えば、入力値のニューロン数は調整期間と調整期間を表現できる最小数は121（入力層の右側も同様、但し結合や増幅となるのなら、この分ニューロンを増やす必要がある）、出力値のニューロン数は11（調整期間を決定できる最小の入力層と同様、結合や増幅なら増加）、出力層から入力層へフィードバックするニューロン数と、中間層のニューロン数は、（決定されている）最適解が導かれるまでの組み合わせの数で決定できる（決定した調整期間の組み合わせが最適解であれば中間層のニューロン数は151、それ以上であり、フィードバックするニューロン数も同様である）。

[0055] 当然、単純な3層型（入力層・中間層・出力層）の、バックプロパゲーションネットワークでも解法は得られるが、出力値の一部をフィードバックさせることによって高次的な情報である前回結果を加味した方が、最適化を向上させられる。

[0056] 最初の学習には多少の時間が必要となるが、一度学習してしまえば、例えば、安定したところに、新たなより優先度の高い調整タスクが加えられた場合、CPUリソースが物理的に割り当てが不可能でない限り、既存の調整タスクのスケジュールを修正すること無く、最適化周期で実行することが可能になる。

[0057] 以上のように、本実施の形態によれば、各調整タスクにおいて、ニューロンを必要数だけ有する最適化機構において、これらのニューロンを結合したニューラルネットワークを用い、過去の（時系列的な）周期調整の情報を有効に用いることにより、実行間隔を容易に最適化することができ、定期調整と実行する調整の調整タスクが実行されるときは、どの調整タスクも調整タスク毎に設定された一定の周期で定期調整が実行できるようにスケジュールの最適化を行うことができる。また、より早くスケジュールの最適化を行うことができる。

[0058] 実施の形態4、以下、実施の形態4を図を参照して説明する。本実施の形態は、スケジュール間に命令受付手段を、スケジュール間に命令実行手段を、そして最適化機能内に、1周期タスクの実行スケジュールの最適化を行ってよいを示す図である。すなわち、1周期の最適化を繰り返し、もしくは1は、ならぬ状態を示す調整可能スケジュールを有することにより、ユーザが希望する調整タスクに対して調整作業を行

(8)

うことができるものである。調整可能スケジュールは、一つの調整タスクに対してだけでなく、全調整タスクに同じ情報を有している。調整可能スケジュールの要素は、調整可能調整タスク数分増えた構造体、または、1ビットに1周期タスクを割り当てた変数でもよい。

[0059] 調整可能スケジュールの値の初期化処理は、システムの初期化に行き、元の情報は、スケジュールで特定の値を入れておく方法でも、調整タスク1 a～1 cの固有情報2 a～2 cの初期の情報をとして追加しておく方法でもよい。

[0060] 以上11は、実施の形態5の調整可能調整タスクの最適化スケジュール方式を示す構成図である。11.1は最適化の判定処理の流れを示すフローチャートである。実施の形態1の11.0の言い、新たに追加した命令実行手段61と、最適化機構61の調整可能スケジュール52が付けられた点である。最適化を行うか行わないかの判定は、11.3に示すフローチャートと同じ処理の流れであり、相違点は11.3のスケジュール104の最適化の判定にて、調整期間が0であるか否か、前回の調整期間が最適調整期間と一致していないかどうかで判定していたのを、11.1.2に示すようにスケジュール131で新たに調整可能スケジュールの最適化を行うことが追加され、調整可能調整タスクの最適化を行うことになる。

[0061] 調整可能スケジュールの修正方法であるが、例えば、カーネル側の最適化機構61のメモリ空間にアクセスが可能であるプログラムを用意し、プログラクショナルにある命令実行手段61は、このプログラムに対して/0命令を実行する形で、所望の調整タスクの調整可能スケジュールを変更できる。

[0062] 以上のように、本実施の形態によれば、スケジュール間に命令受付手段を、プログラクショナル間で命令実行手段を、そして最適化機能内に1周期の決め幅を調整してもよい、もしくはしてはならない状態を示す調整可能スケジュールを有することにより、ユーザが希望する調整タスクもしくは全調整タスクに対して調整作業を行うことができる。また、同時に最適化を中断した調整タスクを取り最適化を実行させることも可能であり、対応に応じて調整タスクの最適化を行うことができる。さらに、定期調整と実行する調整の調整タスクが実行されるときは、どの調整タスクも調整タスク毎に設定された一定の周期で定期調整が実行できるようにスケジュールの最適化を行うことができる。

[0063] 実施の形態6、以下、実施の形態6を図を参照して説明する。本実施の形態は、最適化機能内に最適化を行う時間を、有することにより、ユーザが希望する調整タスクに対して調整作業を行うことができるものである。この最適化を行う時間の変更は、システム全体として存在する場合には、全調整タスクの最適化が行われる時間の制限を、各調整タスク毎に用いる

ことにより、それぞれの調整タスクを個別に制御することが可能である。

[0064] 最適化を行う時間の初期化処理は、スケジュールの初期化に行き、初期化時の値としては、スケジュールで特定の値を入れておく方法（例えば、どの調整タスクも必ず最適化を10000という初期の調整行いたい値の値をすべて1とし、場合によっては、この変数だけに1を2 c内に所望の値ととして追加しておく方法でもよい。

[0065] 以上11は、実施の形態6の調整可能調整タスクの最適化スケジュール方式を示す構成図である。11.1.4は、最適化の判定処理の流れを示すフローチャートであり、11.1.3のフローチャートに、スケジュール141の最適化調整期間の値と、スケジュール142の最適化調整期間が0か、0以外か、が追加されている。なお、最適化時間の値の変更方法は、実施の形態8と同じである。

[0066] 以上のように、本実施の形態によれば、スケジュール間に命令受付手段を、プログラクショナル間で命令実行手段を、そして最適化機能内に1周期調整の時間を、有することにより、ユーザが希望する調整タスクもしくは全調整タスクに対して調整作業を行うことができる。また、所定の調整期間にどの調整タスクも調整タスク毎に設定された一定の周期で定期調整が実行できるようにスケジュールの最適化を行うことができる。

[0067] 実施の形態7、以下、実施の形態7を図を参照して説明する。本実施の形態は、最適化機構と同様のものをプログラクショナルに持つていき、プログラクショナルにも最適化機構と同様のものを配置したものである。11.1.5は、実施の形態7の調整可能調整タスクの最適化スケジュール方式を示す構成図であり、最適化機構6とは同様の処理を行うことができる。AP調整可能機構（シミュレータ）71が追加されている。

[0068] AP調整可能機構71には、カーネル側の最適化機構61の調整可能スケジュール52にある調整可能調整タスク4と、同じくカーネル側の調整可能調整タスク4のスケジュールを保持している。また、カーネル側のスケジュール7 aを保持している。また、カーネル側のスケジュール7 bを保持している。さらに、AP調整可能機構71の情報を保持している。さらに、AP調整可能機構71に対して指示を与えたりするための出力機構70がある。

[0069] この入力機構70では、AP調整可能機構71で行われているシミュレーション情報（各調整

タスクの終結や、現在でのスケジューリング状況の表示物)を面に出し、このシミュレーション情報に対してユーザが各周期タスクもしくは全周期タスクのスケジューリングを最適化するためのパラメータ変更を頻りに試みることが可能になる。例えば、なにか最適化が行われないような状態であることが出力内容から適度で読み取れる、ある周期タスク1の最適時間を広く試みることなどを試すことができる。

[0070] 同様に、動的な動的再処理は、実際に発生したスケジューリング、または実行がされている周期タスクに対して行うことは、当然な構成である場合があるが、数多く複雑な構成は図面であるが、AP最適化機構71(シミュレータ)を介することによって周期タスクの実動時間を変更し、ユーザが理解しやすい(追跡容易しやすい)状況を作り出すことが可能になる。

[0071] 同様に、動的な動的再処理は、実際に発生したスケジューリング、または実行がされている周期タスクに対して行うことは、当然な構成である場合があるが、数多く複雑な構成は図面であるが、AP最適化機構71(シミュレータ)を介することによって周期タスクの実動時間を変更し、ユーザが理解しやすい(追跡容易しやすい)状況を作り出すことが可能になる。

[0072] カネル側の最適化機構6とアプリケーション側のAP最適化機構71との間の通信には、前実施の形態のように命令実行手段51を介して行う。AP最適化機構71には、カネル側のスケジューラ4や周期タスク情報テーブル3、スケジューラ情報5の類似的なものを保持しており、カネル側で行われるスケジューリングと同等の処理を行える。

[0073] 図18は、シミュレーションの処理の流れを示すフローチャートである。図において、ステップS151で現状の周期タスクの情報を獲得する。獲得する際は、命令実行手段51を介することによって、例えば、1の命令を実行し、最適可能スケジュール52及び最適化実行時間61、スケジューラ情報5の状態を保持するメモリ領域を参照する。

[0074] 次に、ステップS152で疑似スケジューラ72がスケジューリングを行う。疑似スケジューラ72では、カネル側のスケジューラ4のうち、周期タスク1のスケジューリングに関する部分のみを模した機能を実行しており、上記で得た情報を基に、どのようなスケジューリングが行われているかを計算する。

[0075] 次に、ステップS153でスケジューリングの状態をユーザが理解できるように画面に出す。表示の形態は、図9や図13のようなもので、周期タスク数及び周期情報の種類が多い場合には、最大の周期時間が複数表示できるものを最大値として表示可能なことが望ましい。彼は、表示画面の一部を選択すると、この部分が拡大されるようにすれば、ユーザ側が必要情報は十分に得られる。

[0076] 次に、ステップS164でユーザはこれら

の出力情報を基にシミュレーション処理を行う。機能として、時間を進めてみてスケジューリングがどのように変化し、最適化するの、特定及び全周期タスクの特定の情報(例えば最適時間や場合には最適時間)を変更できるものがある。例えば、設定の変更を行う際は、シミュレーション作業は停止しているため、パラメータ(各情報)を変更した後は、ステップS155でシミュレーションを再度実行するが終了するかを選択する。再度実行を行う場合にはステップS152に戻り、処理を繰り返す。再度実行を行わずに場合には処理を終了する。

[0077] 以上のように、アプリケーション側にAP最適化機構71及び、入出力機構70を有することによって、ユーザが現状の周期タスクスケジューリングの状況を理解しやすくなるので、シミュレーションが行いやすく、ユーザが周期タスクの初期設定を的確に行うことができる。また、ユーザの指示に従って、どの周期タスクも周期タスク毎に設定された一定の周期で定期処理が実行できるようなスケジューリングの最適化を行うことができる。

[0078] 実施の形態8、以下、実施の形態8を図を参照して説明する。本実施の形態は、アプリケーション側にて最適化の計算を行っておき、最適解が求まるところからこの結果をカネル側の最適化機構の戻し、スケジューリングを行わせるものである。図17は、実施の形態8の複数周期タスクスケジューラ4の最適化スケジューリング方式を示す機能図であり、スケジューラ4の起動停止機構81とスケジューラ内容変更機構82が追加されている。図18は本実施の形態のスケジューリングの処理の流れを示すフローチャートである。

[0079] まず、ステップS161で実行行われているスケジューリングの停止前スケジューラ4の停止を行う。命令の圧迫方法としては、入出力機構70からユーザが命令を送り、AP最適化機構71及び命令実行手段51を介して、カネル側の最適化機構6内のスケジューラ起動停止機構81に送る方法が考えられる。スケジューラ起動停止機構81では、現在スケジューリングされている周期タスク1の実行を一時停止させる。例えば、図17に示される例では、スケジューリング情報を空にしていまい、周期タスクの実行を止める方法が考えられる。この方法では、優先度別にタイムアウトキープがある中で、実行中の周期タスクを完了した後は、このキープの内容を別の領域に格納し、キープ自体は初期化をしよう。

[0080] 次に、ステップS151でカネル側より、シミュレータが必要とする情報をAP最適化機構71へ送るのがあるが、前回のスケジューラ送中のスケジューラ情報を含め、実施の形態7の図16と同様の処理を行う。ステップS161からステップS164のユーザ側からの指示を受けシミュレーションを繰り返す処理は、実施の形態7と同じである。

[0081] 次に、ステップS162でシミュレーションによって得られた情報をカネル側の最適化機構6に送り、ステップS163で再度スケジューリングを開始する。この時、例えば、シミュレーションによって得られた全情報(疑似スケジューラが保持するスケジューリング情報はカネル側のスケジューラ情報5に与えることにより)を送ることにより、即座に最適化されたスケジューリングが実行可能になる。また、スケジューラ情報は更新せず、各周期タスクの情報をテーブルに含められた情報を更新し、最適化を行う時から処理を再度スタートすることも可能である。

[0082] 以上のように、アプリケーション側に最適化機構及び、入出力機構、スケジューラ起動停止機構、スケジューラ内容変更機構を有することによって、ユーザが現状の周期タスクスケジューリングの状況を理解しやすくなるので、シミュレーションが行いやすく、このシミュレーション結果をカネル側のスケジューリングに送ることによって、どの周期タスクも周期タスク毎に設定された一定の周期で定期処理が実行できるようなスケジューリングの最適化を行うことができる。

[0083] 実施の形態9、以下、実施の形態9を図を参照して説明する。本実施の形態は、アプリケーション側にて最適化の計算を行っておき、最適解が求まるところからこの結果をカネル側の最適化機構の戻し、スケジューリングを行わせる際に、新たに周期タスクの追加を行った場合のシミュレーション及びこの結果をカネル側へ戻すのを可能とするものである。

[0084] 図19は、実施の形態9の複数周期タスクスケジューラ4の最適化スケジューリング方式を示す機能図であり、新規タスク追加機構91が追加されている。例えば、既存の周期タスク1を新規に追加する場合には、ユーザが実行の入出力機構70を介して、AP最適化機構71内の新規タスク追加機構91に周期タスク12の具体的なアドレスと固有情報2を渡される。具体的なアドレスは、シミュレーションを行う際、利用しないが、シミュレーションの結果をカネル側に渡し、実際に実行する際に必要になる。

[0085] 本実施の形態の処理は、シミュレータの側に新規タスクを追加する部分と、シミュレータ後の情報をカネル側に送る処理とに分かれる。図20は、新規タスクが追加された時の処理の流れを示すフローチャートである。図において基本的な処理の流れは、実施の形態8の図19におけるステップS161、ステップS151～S153の部分と一緒である。唯一の違いは、ステップS154でユーザ側から新規タスクの追加の指示を受けると、新規周期タスクの情報(具体的なアドレス、詳細情報)と、新規タスク追加機構91に渡し、ステップS170で疑似スケジューラ情報の更新を行う点である。

[0086] 新規タスク追加機構91が起動されると(本機構が起動される時というのは、入出力機構70からの入力待ち状態、つまり、疑似スケジューラが実行していない状態)、新規タスクの詳細情報を疑似スケジューラ72に渡す。疑似スケジューラ72は、前述したように、カネル側のスケジューラ4における周期タスクのスケジューリング機構を模したものであり、疑似スケジューラ情報76を基にスケジューリングを行っている。

[0087] 新規タスク追加機構91は、新規の周期タスク12の固有情報2に基づき、疑似スケジューラ情報76に追加を行う。例えば、図中のスケジューラ情報5に示されるように優先度別のタイムアウトキープでスケジューリングが管理されている場合には新規の周期タスク12の優先度を参照し、このキープに追加を行う。

[0088] 次に、カネル側へ新情報を更新する時の動作について説明する。基本的には、実施の形態8と同様であるが、新規の周期タスク12が追加されているので、周期タスク12の実行を行えるように必要な情報をカネル側のスケジューラ本体4に伝えなければならぬ。これは、スケジューラ本体が周期タスクを実行するには、周期タスク情報テーブル3を参照し、実行アドレスの情報を得るので、今回新規に追加した周期タスクもこの周期タスク情報テーブル3へ追加しなければならないことによるのである。

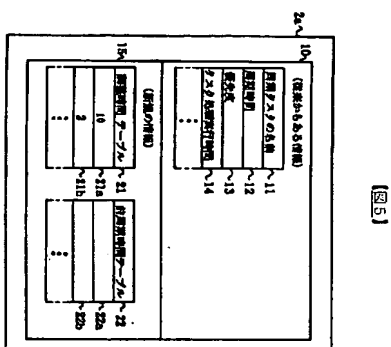
[0089] これらの処理は、実施の形態8にて使用したスケジューラ内容更新機構82にて、周期タスク情報テーブル用のメモリ領域を確保し、このメモリ領域に命令実行手段51を介してAP最適化機構71内の新規周期タスク追加機構91が保持する周期タスクのアドレス等の情報を格納し、スケジューラ本体4が参照できるようにする方が考えられる。また、新たに追加される周期タスクを見逃して、初期化処理時に周期タスク情報テーブル3の領域を大きく目に確保しておく方法も考えられる。

[0090] 以上のように、アプリケーション側のAP最適化機構71内に新規周期タスク追加機構91を有することによって、ユーザが既存の周期タスクスケジューリングの状況を理解しやすくし、シミュレーションを行い、かつ、新規の周期タスク12を追加し、どのようなスケジューリングが行われるのかを予め確認、この結果が良好であればこの結果をカネル側のスケジューリングに送ることによって、新規周期タスクが追加されたときにもどの周期タスクも周期タスク毎に設定された一定の周期で定期処理が実行できるようなスケジューリングの最適化を行うことができる。

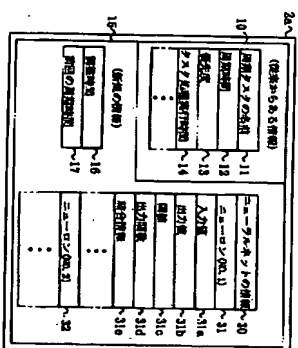
[0091]

[発明の効果] この発明は、以上説明したように構成されているので、以下に示すような効果を奏する。

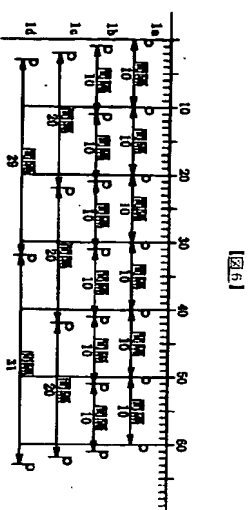
[0092] 第1の発明では、定期処理を実行する度



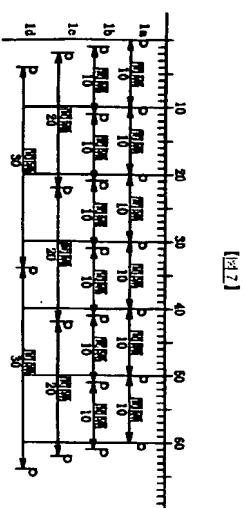
【圖5】



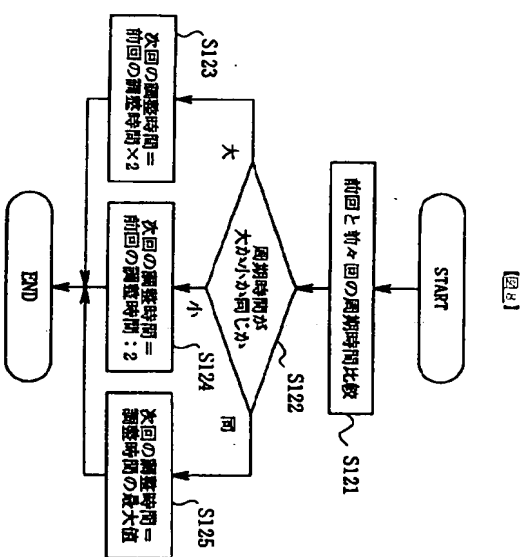
【例 3】



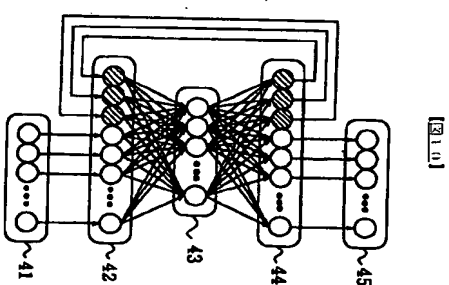
【58】



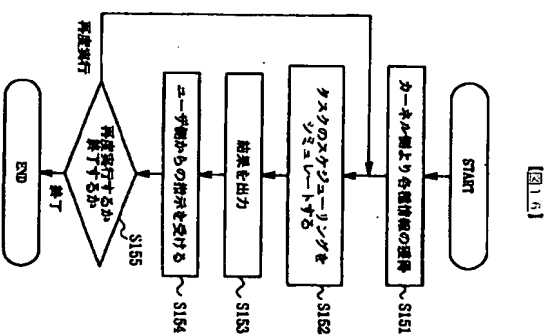
【图 7】



【例】

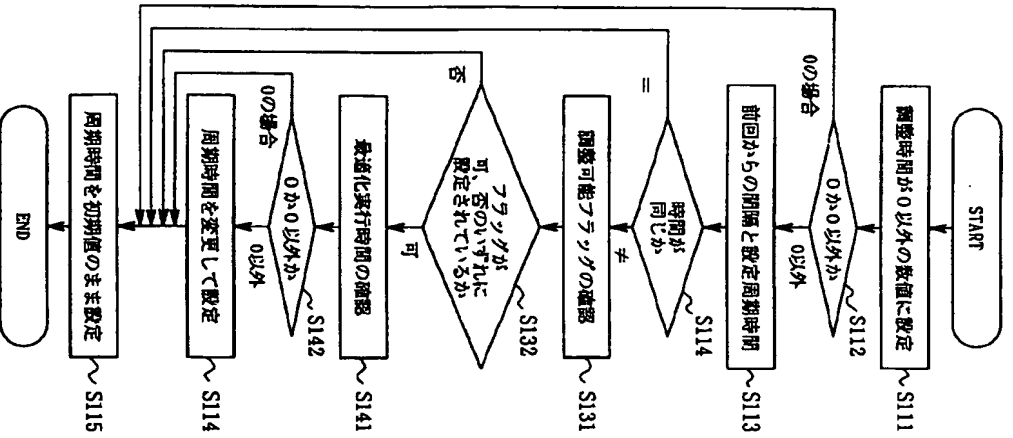


【圖 1 (1)】

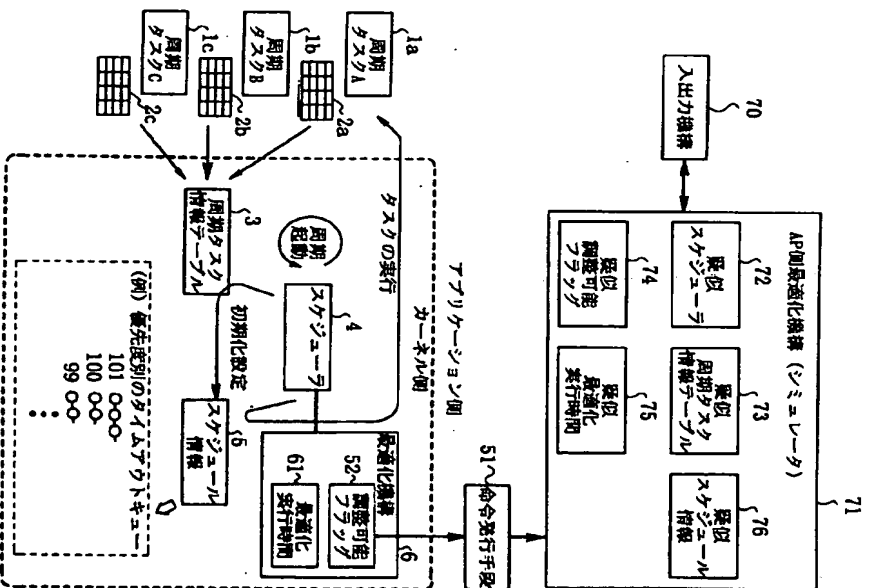


【圖 15】

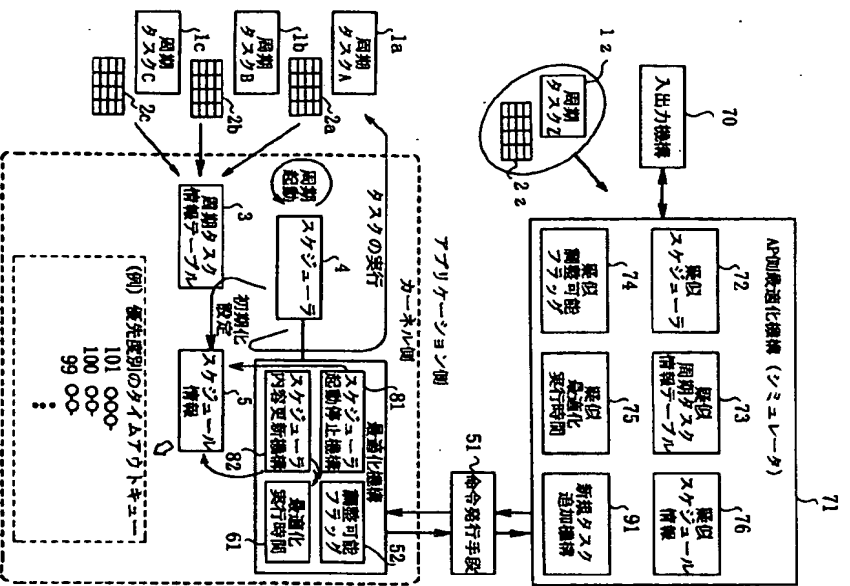
【図14】



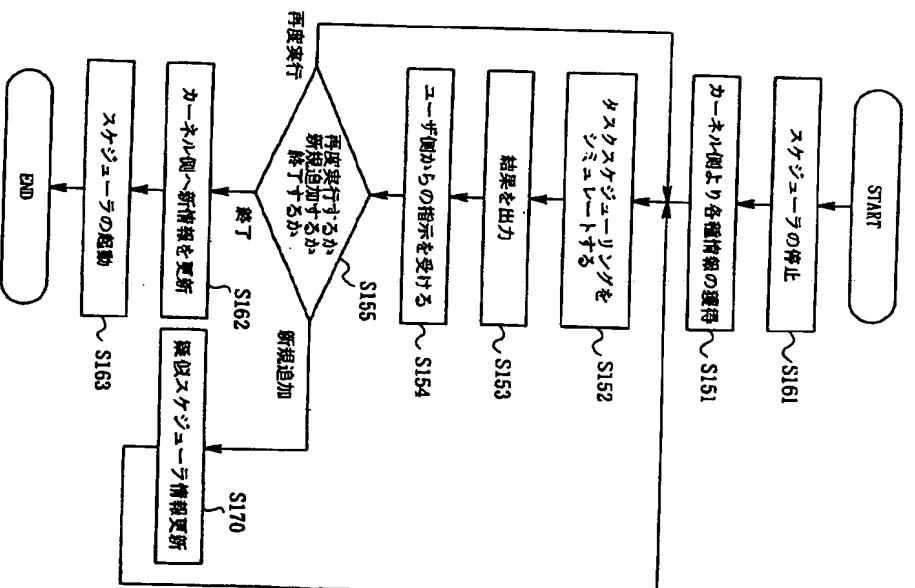
【図15】



【図19】



【図20】



THIS PAGE BLANK (USPTO)